



ELSEVIER

Discrete Applied Mathematics 96–97 (1999) 247–257

DISCRETE
APPLIED
MATHEMATICS

[Metadata, citation and](#)

her Connector

Accuracy of techniques for the logical analysis of data

Martin Anthony

*Department of Statistics and Mathematics, London School of Economics, Houghton Street,
London WC2A2AE, UK*

Received 30 January 1996; revised 20 December 1998; accepted 1 February 1999

Abstract

We analyse the generalisation accuracy of standard techniques for the ‘logical analysis of data’, within a probabilistic framework. © 1999 Elsevier Science B.V. All rights reserved.

1. Introduction

There has recently been some interest in operations research in ‘the logical analysis of data’ [5] (which we shall often abbreviate to LAD). The main emphasis so far has been on finding convenient and informative explanations of data by means of Boolean functions. The accuracy of these explanations on classifying unseen data has been estimated experimentally, and the results are impressive; certainly, the performance of the techniques is comparable with those of popular machine learning techniques. In this paper, we apply techniques from the probabilistic analysis of machine learning – as discussed in the books [2,8,9], for example – to analyse theoretically the accuracy of the LAD techniques.

In Section 2, we introduce the basic notations and describe important classes of Boolean functions. In Section 3 we describe the standard LAD techniques and in Section 4 we analyse their accuracy.

2. Boolean function classes

A Boolean function (of n variables) is usually taken to be a function from $\{0,1\}^n$ to $\{0,1\}$. Sometimes it is useful to regard a Boolean function as a mapping from $\{-1,1\}^n$ to $\{0,1\}$. When taking the first approach, we say that we are using the *standard* convention, and we shall refer to the latter as the *nonstandard* convention. Transforming

E-mail address: m.anthony@lse.ac.uk (M. Anthony)

from standard to nonstandard conventions is simple, via the transformation $y \rightarrow 2y - 1$, mapping 0 to -1 and 1 to 1.

We shall not give here a detailed exposition of Boolean functions and formulae; full details may be found in many texts. Recall that any Boolean function can be expressed by a *disjunctive normal formula* (or DNF), using *literals* $u_1, u_2, \dots, u_n, \bar{u}_1, \dots, \bar{u}_n$, where the \bar{u}_i are known as *negated literals*. A disjunctive normal formula is one of the form

$$T_1 \vee T_2 \vee \dots \vee T_k,$$

where each T_l is a *term* of the form

$$T_l = \left(\bigwedge_{i \in P} u_i \right) \wedge \left(\bigwedge_{j \in N} \bar{u}_j \right),$$

for some disjoint subsets P, N of $\{1, 2, \dots, n\}$. A Boolean function is said to be an l -DNF if it has a disjunctive normal formula in which, for each term, the number of literals ($|P \cup N|$) is at most l ; it is said to be a k -term- l -DNF if there is such a formula in which, furthermore, the number of terms T_i is at most k .

We now describe the polynomial threshold functions (of a given degree), a useful class of Boolean functions. Let $[n]^{(d)}$ denote the set of all subsets of at most d objects from $[n] = \{1, 2, \dots, n\}$. For any $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$, x_S shall denote the product of the x_i for $i \in S$. For example, $x_{\{1, 2, 3\}} = x_1 x_2 x_3$. When $S = \emptyset$, the empty set, we interpret x_S as the constant 1. With this notation, a Boolean function f defined on $\{0, 1\}^n$ is a *polynomial threshold function of degree d* if there are real numbers w_S , one for each $S \in [n]^{(d)}$, such that

$$f(x) = 1 \Leftrightarrow \sum_{S \in [n]^{(d)}} w_S x_S > 0.$$

This may be written

$$f(x) = \text{sgn} \left(\sum_{S \in [n]^{(d)}} w_S x_S \right),$$

where the *sign function* sgn is such that $\text{sgn}(x) = 1$ if $x > 0$ and $\text{sgn}(x) = 0$ if $x \leq 0$. The set of polynomial threshold functions on $\{0, 1\}^n$ of degree d will be denoted by $\mathcal{P}(n, d)$. The class $\mathcal{P}(n, 1)$ is usually known simply as the set of *threshold functions* on $\{0, 1\}^n$. It is easy to see that any l -DNF f on $\{0, 1\}^n$ is in $\mathcal{P}(n, l)$, as follows. Given a term $T_j = u_{i_1} u_{i_2} \dots u_{i_r} \bar{u}_{j_1} \bar{u}_{j_2} \dots \bar{u}_{j_s}$ of the DNF, we form the expression

$$A_j = x_{i_1} x_{i_2} \dots x_{i_r} (1 - x_{j_1}) (1 - x_{j_2}) \dots (1 - x_{j_s}).$$

We do this for each term T_1, T_2, \dots, T_k and expand the algebraic expression $A_1 + A_2 + \dots + A_k$ according to the normal rules of algebra, until we obtain a linear combination of the form $\sum_{S \in [n]^{(l)}} w_S x_S$. Then, since $f(x) = 1$ if and only if $A_1 + A_2 + \dots + A_k > 0$,

it follows that

$$f(x) = \operatorname{sgn} \left(\sum_{S \in [n]^{(l)}} w_S x_S \right),$$

so $f \in \mathcal{P}(n, l)$.

We shall find it useful to specify certain subclasses of $\mathcal{P}(n, d)$. First, we define the class $\mathcal{B}(n, d)$ of *binary-weight* polynomial threshold functions to be the functions in $\mathcal{P}(n, d)$ for which the weights w_S all belong to $\{-1, 0, 1\}$ for $S \neq \emptyset$, and for which $w_\emptyset \in \mathbb{N}$ (where \mathbb{N} is the set of natural numbers). Next, for $1 \leq j \leq \sum_{i=0}^d \binom{n}{i}$, define $\mathcal{P}_j(n, d)$ to be the set of all functions in $\mathcal{P}(n, d)$ with at most j of the weights w_S non-zero for $S \neq \emptyset$; thus a function is in $\mathcal{P}_j(n, d)$ if and only if there are non-empty subsets S_1, S_2, \dots, S_j of $\{1, 2, \dots, n\}$, each of cardinality at most d , and constants $w_0, w_1, w_2, \dots, w_j$ such that

$$f(x) = 1 \Leftrightarrow w_0 + \sum_{i=1}^j w_i x_{S_i} > 0.$$

We shall say that the functions in $\mathcal{P}_j(n, d)$ *involve at most j product terms*. In an analogous way we can define $\mathcal{B}_j(n, d)$, the class of binary-weight polynomial threshold functions involving at most j terms w_S where $S \neq \emptyset$, and which have $w_\emptyset \in \{0, 1, \dots, j-1\}$. We have remarked that any l -DNF function lies in $\mathcal{P}(n, l)$; in fact, it lies in the subclass $\mathcal{B}(n, l)$. When using the standard convention for Boolean functions, it is not generally true that a k -term- l -DNF lies in $\mathcal{B}_k(n, l)$; all that can be said is that it lies in $\mathcal{B}(n, l)$; however, if we use the nonstandard convention, it is the case that $f \in \mathcal{P}_k(n, l)$. For, instead of replacing a negated literal \bar{u}_i in a term by the algebraic expression $1 - x_i$, we replace it simply by $-x_i$; it is clear that the product terms of the resulting polynomial threshold function are in one-to-one correspondence with the terms of the DNF formula and that they have precisely the same degree. (We take $w_\emptyset = j - 1$ where $j \leq k$ is the number of terms in the DNF.)

3. Logical analysis of data

In this section, we briefly describe the basic aims and techniques of logical analysis of data; a more complete discussion may be found in [5].

In LAD, one is given some elements of $\{0, 1\}^n$, classified according to some *hidden function* t : a given $x \in \{0, 1\}^n$ in the data set is classified as *positive* if $t(x) = 1$ and *negative* if $t(x) = 0$. The data points, together with the positive/negative classifications will be denoted D . An *extension* of D is a Boolean function f such that f agrees with D ; that is, if x is one of the data points given in D then $f(x) = 1$ if and only if x is classified as positive in D . The aim is to find an extension of f which can be described very ‘simply’; for example, a k -term- l -DNF, where k, l are small. In a sense, the extension ‘explains’ the given data and it is to be hoped that it generalises well

to other data points, so far unseen. That is, we should like it to be the case that for most $y \in \{0, 1\}^n$ which are not in D , the extension f classifies y correctly, by which we mean $f(y) = t(y)$. There are clearly very many extensions of a given data set. We shall analyse the performance of standard LAD methods.

In the standard method described in [5], a DNF is produced. First, a *support set* of variables is found. This is a set $S = \{i_1, i_2, \dots, i_s\}$ such that no positive data point agrees with a negative data point in the coordinates i_1, i_2, \dots, i_s . If S is a support set then there is some extension of D which depends only on the literals u_i, \bar{u}_i for $i \in S$ (and conversely). In the technique described in [5], a small support set is found by solving a set-covering problem derived from the data set D . Once a support set has been found, one then looks for *patterns*. These are conjunctions of literals which are satisfied by at least one positive example in D but by no negative example. We then take as the extension f the disjunction of a set of patterns which together cover all positive examples (that is, which are such that each positive example satisfies some pattern). Suppose that the chosen support set has cardinality s , that each pattern is a conjunction of at most $d \leq s$ literals, and that the number of patterns is P ; then the resulting formula for the extension is a P -term- d -DNF formula.

There are some variants on this method. It is also possible to make use of *negative patterns*. A negative pattern is a conjunction of literals which is satisfied by at least one negative example and by no positive example. Suppose that T_1, T_2, \dots, T_q are patterns covering all positive examples in D and that T'_1, T'_2, \dots, T'_r are negative patterns covering all negative examples in D . Then the function

$$f = \text{sgn} \left(\sum_{i=1}^q T_i - \sum_{j=1}^r T'_j \right)$$

is easily seen to be an extension of D . If each pattern and negative pattern is a conjunction of at most d literals, then the resulting extension lies in $\mathcal{B}_{q+r}(n, d)$.

There might be some advantage in ‘weighting’ the patterns, assigning positive weights to the patterns and negative weights to the negative patterns; that is, we take as extension a function of the form

$$f = \text{sgn} \left(\sum_{i=1}^q w_i T_i - \sum_{j=1}^r w'_j T'_j \right),$$

where the w_i, w'_j are positive. If we use weights in this manner, it may be easier to ‘update’ the extension should we subsequently be presented with more classified data points. Note that the total number of patterns used by the LAD method described above is certainly no more than m , the number of data points.

4. Generalisation from random data

Given an extension of a fairly large data set determined by LAD techniques, it is important to know how well it would classify further data points. In the absence of

any other information about the hidden function or the extension, we cannot guarantee anything: it is quite possible that, although the extension classifies correctly the data already seen, it disagrees with the hidden function's classification on all other possible data. Often, one assumes that the data is described by a hidden Boolean function of a *particular type*. Moreover, the LAD techniques described in the previous section produce as extensions of the data particular types of DNF or, more generally, polynomial threshold functions. There are therefore biases built in to the techniques: one bias is the assumption of a particular type of hidden function and the other is the use of a particular type of extension. The two are often related: one might use a certain type of extension because it is thought to be of the same form as the hidden function. But nothing might be known about the hidden function. Then, if a particular type of simple extension to a fairly large data set can be found (for example, one with small patterns), then, even if we cannot be sure that the hidden function is so simple, we might still expect, given the success of the simple extension in explaining the large data set, that this extension will perform well on 'most' unseen data. (This is, in some senses, an instance of the 'Occam's razor' principle: we trust a simple explanation of the data.) Issues such as these have been well-studied in 'computational learning theory' and 'statistical learning theory'. To formalise the ideas somewhat, we assume that the types of extension which can be produced all belong to a particular class, H , of functions, known as the *hypothesis space*. The choice of hypothesis space might reflect either our belief about the type of hidden function, or our intention only to accept simple types of explanation of the data. For instance, we might have as hypothesis space the Boolean functions which arise from using at most 15 patterns, each involving at most 5 literals; that is, the hypothesis space consists of 15-term-5-DNF functions. Alternatively, H might be some class $\mathcal{P}_k(n, d)$, consisting of polynomial threshold functions of degree d , each involving at most k product terms.

We have seen (in Section 3) that the basic LAD technique produces an extension belonging to the class of P -term d -DNF Boolean functions, where P is the number of patterns used and d the maximum number of literals in any of the patterns. For the extended LAD techniques, the algorithms give an extension which lies in a class $\mathcal{P}_P(n, d)$ of polynomial threshold functions. We shall apply some probabilistic techniques to analyse the performance of LAD algorithms on random data. These methods have been used in learning theory (see [2,4,9]) and originated in the work of Vapnik and Chervonenkis [10]. Following the PAC model of computational learning theory, we assume that the data points are generated randomly according to a fixed probability distribution μ on $\{0, 1\}^n$ and that they are classified by some hidden function t . Thus, if there are m data points in D , then we may regard the data points as a vector in $(\{0, 1\}^n)^m$, drawn randomly according to the product probability distribution μ^m . Given any extension f of a data set D (which it will be presumed belongs to some hypothesis space), we measure how well f performs on further examples by means of its *error*

$$\text{er}(f) = \mu(\{x \in \{0, 1\}^n : f(x) \neq t(x)\}),$$

which is the probability that f incorrectly classifies an $x \in \{0, 1\}^n$ drawn randomly according to μ . (Note that such a random x may be one of the data points of D .)

Our main results are the following. The first gives a bound which is useful in measuring the accuracy of the standard LAD methods and the second gives a bound which is useful in the more general methods in which patterns are weighted. It should be noted that the first result, for example, requires P and d to be fixed in advance, to enable a concrete hypothesis space to be specified. The second result similarly requires the precise hypothesis space to be specified in advance. This over-prescriptiveness is problematic: one would not necessarily know, before the event, that the LAD algorithm is likely to produce a 15-term-5-DNF extension, for example. Nonetheless, we shall see in due course that the following two results can be modified to remove this artificial degree of specification.

Theorem 4.1. *Suppose that d and P are fixed positive integers and that D is a data set of m points, each generated at random according to a fixed probability distribution on $\{0, 1\}^n$, where $n \geq 2$. Let δ be a positive number less than one. Then the following holds with probability at least $1 - \delta$: Suppose that f is an extension of D and that f is either a P -term- d -DNF or a binary-weight polynomial threshold function in $\mathcal{B}_P(n, d)$. Then the error of f is less than*

$$\frac{Pd \ln n + (2P + 1) \ln 2 + \ln(P/\delta)}{m},$$

for $n \geq 2$.

Theorem 4.2. *Suppose that d, P, m and n are fixed positive integers, where $n \geq 2$ and $P \leq 2m$, and suppose that D is a data set of m points, each generated at random according to a fixed probability distribution on $\{0, 1\}^n$. Let δ be a positive number less than one. Then the following holds with probability at least $1 - \delta$: Suppose that f is an extension of D and that f is a polynomial threshold function in $\mathcal{P}_P(n, d)$. Then the error of f is less than*

$$\frac{2Pd \ln n + 2(P + 1) \ln m + (4P + 6) \ln 2 + 2 \ln(1/\delta)}{m \ln 2}.$$

Note that the restriction $P \leq 2m$ is certainly true for the LAD techniques described in the previous section. (Indeed, we have $P \leq m$.)

The proofs of these results follow fairly easily from two bounds from probability theory and computational learning theory.

Given $\mathbf{x} = (x_1, x_2, \dots, x_m) \in (\{0, 1\}^n)^m$ and a hidden Boolean function t , we denote by $D(\mathbf{x}, t)$ the data set consisting of the x_i together with their classifications by t . Given a data set D , we shall find it convenient to write $f \succ D$ to mean that f is an extension of D . The first bound is standard and can be found in [4], for example: if H is a set of Boolean functions on $\{0, 1\}^n$, $t : \{0, 1\}^n \rightarrow \{0, 1\}$ and μ is a probability distribution on $\{0, 1\}^n$, for any positive integer m and any $\varepsilon \in (0, 1)$,

$$\mu^m(\{\mathbf{x} : \exists f \in H \text{ such that } f \succ D(\mathbf{x}, t) \text{ and } \text{er}(f) \geq \varepsilon\}) < |H| \exp(-\varepsilon m).$$

Proof of Theorem 4.1. If we use the nonstandard convention for Boolean functions, then the class of P -term- d -DNF functions is a subclass of the class of binary-weight polynomial threshold functions $\mathcal{B}_P(n, d)$. It follows that the number of such DNF functions is bounded by $|\mathcal{B}_P(n, d)|$. Therefore, to bound $|H|$ in either case, we bound the cardinality of $\mathcal{B}_P(n, d)$. Recall that $f \in \mathcal{B}_P(n, d)$ if for some $j \leq P$ there are non-empty subsets S_1, S_2, \dots, S_j of $\{1, 2, \dots, n\}$, each of cardinality at most d , and constants $w_1, w_2, \dots, w_j \in \{-1, 1\}$ and $w_0 \in \{0, 1, \dots, P-1\}$ such that

$$f(x) = 1 \Leftrightarrow w_+ \sum_{i=1}^j w_i x_{S_i} > 0.$$

The number of possible such x_S is

$$N = \binom{n}{\leq d} = \sum_{i=0}^d \binom{n}{i}$$

which, for $n \geq 2$, is at most $2n^d$. To count the number of functions in $\mathcal{B}_P(n, d)$, we observe that, given the product terms which such an f involves, there are two choices for the weight assigned to each (either -1 or 1). Furthermore, there are P choices for w_0 . Therefore,

$$\begin{aligned} |\mathcal{B}_P(n, d)| &\leq P \sum_{j=0}^P \binom{N}{j} 2^j \\ &< P 2^P \sum_{j=0}^P \binom{N}{j} \\ &\leq P 2^P (2N^P) \\ &\leq P 2^{P+1} (2n^d)^P \\ &= P 2^{2P+1} n^{Pd}. \end{aligned}$$

Suppose the distribution generating the data is μ . Then, by the result quoted above, with H equal either to the class of P -term- d -DNF or to $\mathcal{B}_P(n, d)$,

$$\begin{aligned} \mu^m(\{x: \exists f \in H \text{ s.t. } f \succ D(x, t), \text{ er}(f) \geq \varepsilon\}) &< |H| \exp(-\varepsilon m) \\ &< P 2^{2P+1} n^{Pd} \exp(-\varepsilon m). \end{aligned}$$

If

$$\varepsilon \geq \frac{dP \ln n + (2P+1) \ln 2 + \ln(P/\delta)}{m}$$

then this probability is less than δ . The result follows. \square

To prove the second theorem we use the following result from [4], which follows [8]. (The result has subsequently been improved upon; see [6,3]. Using these improvements would result in an error bound a constant factor smaller than that given in

Theorem 4.2.) With the notation as above, the bound stated that for any positive integer $m \geq 8/\varepsilon$ and any $\varepsilon \in (0, 1)$,

$$\mu^m(\{x: \exists f \in H \text{ such that } f \succ D(x, t) \text{ and } \text{er}(f) \geq \varepsilon\}) < 2\Pi_H(2m)2^{-\varepsilon m/2},$$

where for a positive integer k , $\Pi_H(k)$ is the maximum cardinality of H restricted to any k -subset of $\{0, 1\}^n$.

Proof of Theorem 4.2. Let m and δ be given and let

$$\varepsilon = \frac{2Pd \ln n + 2(P+1) \ln m + (4P+6) \ln 2 + 2 \ln(1/\delta)}{m \ln 2}.$$

Given that $n \geq 2$, we have $m \geq 8/\varepsilon$. Therefore,

$$\mu^m(\{x: \exists f \in H \text{ such that } f \succ D(x, t) \text{ and } \text{er}(f) \geq \varepsilon\}) < 2\Pi_H(2m)2^{-\varepsilon m/2},$$

where $H = \mathcal{P}_P(n, d)$. We bound $\Pi_H(k)$ as follows. As usual, let $[n]^{(d)}$ be the set of all subsets of $\{1, 2, \dots, n\}$ of cardinality at most d and, for $\mathcal{R} \subseteq [n]^{(d)}$, let $H^{\mathcal{R}}$ be the set of polynomial threshold functions of the form

$$\text{sgn}\left(\sum_{S \in \mathcal{R}} w_S x_S\right),$$

where the w_S are real numbers; that is, those which involve the product terms x_S for $S \in \mathcal{R}$. Then

$$H = \bigcup_{\mathcal{R} \subseteq [n]^{(d)}, |\mathcal{R}| \leq P} H^{\mathcal{R}}.$$

For a subset C of $\{0, 1\}^n$, let $H|_C$ denote the restriction of H to domain C . Then, for any subset C of $\{0, 1\}^n$, of cardinality k ,

$$\begin{aligned} |H|_C| &= \left| \bigcup_{\mathcal{R} \subseteq [n]^{(d)}, |\mathcal{R}| \leq P} H^{\mathcal{R}}|_C \right| \\ &\leq \sum_{\mathcal{R} \subseteq [n]^{(d)}, |\mathcal{R}| \leq P} |H^{\mathcal{R}}|_C| \\ &\leq \sum_{\mathcal{R} \subseteq [n]^{(d)}, |\mathcal{R}| \leq P} \Pi_{H^{\mathcal{R}}}(k), \end{aligned}$$

from which it follows that

$$\Pi_H(k) \leq \sum_{\mathcal{R} \subseteq [n]^{(d)}, |\mathcal{R}| \leq P} \Pi_{H^{\mathcal{R}}}(k).$$

In a manner similar to that used in the previous proof, we can show that the number of such subsets \mathcal{R} is at most $2^{P+1} n^{Pd}$. We now fix $\mathcal{R} \subseteq [n]^{(d)}$, of cardinality $r \leq P$, and bound $\Pi_{H^{\mathcal{R}}}(k)$. In order to do this, we make use of the theory of the *Vapnik–Chervonenkis dimension*. This was introduced in [10] and has been used extensively in computational learning theory; see [2,4,9], for example. Given a set G of functions

from a (not necessarily finite) set X to $\{0, 1\}$, the *VC-dimension* of G , $\text{VCdim}(G)$, is defined to be the largest integer k such that for some set C of cardinality k , $|G|_C = 2^k$. (The VC-dimension is infinite if there is no bound on the cardinality of such sets C .) From Sauer's inequality [7], if $m \geq k \geq 1$, $\Pi_G(k) \leq k^{\text{VCdim}(G)+1}$. We remark that when G is a set of Boolean functions defined on $\{0, 1\}^n$, $\text{VCdim}(G) \leq \log_2 |G|$; furthermore, if $\text{VCdim}(G) \geq 1$, then $\text{VCdim}(G) \geq \log_2 |G|/n$. Thus there is at most a factor of n difference between the VC-dimension of a finite class and the binary logarithm of its cardinality. This gap is real; for many classes G , the VC-dimension of G is of order $\log_2 |G|/n$. (It is for this reason that the present proof does not simply bound the cardinality of $\mathcal{P}_P(n, d)$ and then apply the same bound as used for Theorem 4.1.) Returning to the problem at hand, it can be shown (see [1], for example) that the VC-dimension of $H^{\mathcal{R}}$ is $|\mathcal{R}| = r$, so, for $k \geq r$,

$$\Pi_{H^{\mathcal{R}}}(k) \leq k^{r+1} \leq k^{P+1}.$$

Hence,

$$\Pi_H(k) \leq \sum_{\mathcal{R} \subseteq [n]^{(d)}, |\mathcal{R}| \leq P} \Pi_{H^{\mathcal{R}}}(k) \leq 2^{P+1} n^{Pd} k^{P+1},$$

for $k \geq P$. Therefore (since $m \geq 8/\varepsilon$ and $2m \geq P$),

$$\begin{aligned} \mu^m(\{\mathbf{x}: \exists f \in H \text{ s.t. } f \succ D(\mathbf{x}, t) \text{ and } \text{er}(f) \geq \varepsilon\}) &< 2\Pi_H(2m)2^{-\varepsilon m/2} \\ &< 2^{2P+3} n^{Pd} m^{P+1} 2^{-\varepsilon m/2}. \end{aligned}$$

It follows that, with ε as given, this probability is less than δ . The result follows. \square

As noted earlier, Theorems 4.1 and 4.2 both require that P and d be specified in advance, a degree of precriptiveness which seems artificial. However, it is possible to move on from these results to obtain more practically useful results, in which the *type* of hypothesis space is pre-specified, but not the precise one of that type. Specifically, we can obtain results applicable to the situation in which LAD techniques are applied, and P and d are defined by the output of the LAD algorithm, rather than being prescribed before the LAD algorithm is run: thus (for the first result below), the only assumption we need make is that the extensions belong to the class of P -term-DNF Boolean functions, *for some P and d* rather than for *given, fixed P and d* .

Theorem 4.3. *Suppose that D is a data set of m points, each generated at random according to a fixed probability distribution on $\{0, 1\}^n$, where $n \geq 2$. Let δ be a positive number less than one. Then the following holds with probability at least $1 - \delta$: for any $d, P \geq 1$, if f is any extension of D which is either a P -term- d -DNF or a binary-weight polynomial threshold function in $\mathcal{B}_P(n, d)$, then the error of f is less than*

$$\frac{Pd \ln n + (2P + 1) \ln 2 + \ln(P/\delta) + (d + P) \ln 2}{m},$$

for $n \geq 2$.

Theorem 4.4. Suppose that D is a data set of m points, each generated at random according to a fixed probability distribution on $\{0, 1\}^n$. Let δ be a positive number less than one. Then the following holds with probability at least $1 - \delta$: for any $d, P \geq 1$ with $P \leq 2m$, if f is an extension of D which is a polynomial threshold function in $\mathcal{P}_P(n, d)$, then the error of f is less than

$$\frac{2Pd \ln n + 2(P+1) \ln m + (4P+6) \ln 2 + 2 \ln(1/\delta) + (d+P) \ln 2}{m \ln 2}.$$

Thus, these theorems are similar to Theorems 4.1 and 4.2, but P and d are *not* specified in advance. In Theorems 4.3 and 4.4, the error bounds are slightly larger (with an additional $(d+P) \ln 2$ in the numerators) than the error bounds of the earlier theorems, but this is arguably a price worth paying for more useful results.

We shall prove Theorem 4.4. The proof if Theorem 4.3 is similar and is omitted.

Proof of Theorem 4.4. Let $\delta \in (0, 1)$ be fixed, and let the notation be as above. For $d, P \geq 1$, let $p_{d,P}$ be the probability that there is some $f \in \mathcal{P}_P(n, d)$ which is an extension of the data set D and which satisfies $\text{er}(f) > \varepsilon(m, \delta, d, P)$ where $\varepsilon(m, \delta, d, P)$ is

$$\frac{2Pd \ln n + 2(P+1) \ln m + (4P+6) \ln 2 + 2 \ln(1/\delta) + (d+P) \ln 2}{m \ln 2}.$$

On using $\delta/2^{d+P}$ in place of δ , Theorem 4.4 shows that $p_{d,P} < \delta/2^{d+P}$ for $P \leq 2m$. Now, the probability that for *some* $d, P \geq 1$ with $P \leq 2m$, we can find an extension f of D such that $f \in \mathcal{P}_P(n, d)$ and $\text{er}(f) > \varepsilon(m, \delta, d, P)$ is no more than

$$\sum_{d=1}^{\infty} \sum_{P=1}^{2m} p_{d,P} < \sum_{d=1}^{\infty} \sum_{P=1}^{\infty} \frac{\delta}{2^{d+P}} = \delta.$$

The result follows. \square

It is instructive to see (in rough terms) what these results say about the accuracy of the techniques when the underlying distribution is uniform. Consider the standard LAD techniques which produce a DNF or a polynomial threshold function by determining patterns. Theorems 4.3 and 4.4 can be applied to give performance guarantees. For simplicity, let us take $\delta = 0.1$ and suppose $n \geq 9$. Suppose a random data set D of m points is drawn, each point being equally likely, and that the algorithm uses P patterns, each involving at most d literals (where $P \geq 3$ and $d \geq 4$). Then the error bound given in Theorem 4.3 is at most $2Pd \ln n/m$ and so, with probability at least 0.9, the LAD technique results in an extension which has error at most $2Pd \ln n/m$ which, since the distribution is uniform, means that f agrees with the hidden function on at least a fraction

$$1 - \frac{2Pd \ln n}{m}$$

of all 2^n possible data points in $\{0, 1\}^n$. A similar statement, based on Theorem 4.2, can be made for more general techniques using weighted patterns.

Acknowledgements

This work was carried out primarily when I visited RUTCOR and DIMACS, Rutgers University. I am very grateful to Peter Hammer and Ilya Muchnik for a number of stimulating discussions and to anonymous referees for their comments.

References

- [1] M. Anthony, Classification by polynomial surfaces, *Discrete Appl. Math.* 61 (1995) 91–103.
- [2] M. Anthony, N. Biggs, *Computational Learning Theory: An Introduction*, Cambridge Tracts in Theoretical Computer Science, vol. 30, Cambridge University Press, Cambridge, UK, 1992.
- [3] M. Anthony, N. Biggs, J. Shawe-Taylor, The learnability of formal concepts, *Proceedings of the third Annual Workshop on Comput. Learning Theory*, Morgan Kaufmann, San Mateo, CA, 1990, pp. 246–257.
- [4] A. Blumer, A. Ehrenfeucht, D. Haussler, M.K. Warmuth, Learnability and the Vapnik–Chervonenkis dimension, *J. ACM* 36 (4) (1989) 929–965.
- [5] Y. Crama, P.L. Hammer, T. Ibaraki, Cause-effect relationships and partially defined Boolean functions, *Ann. Oper. Res.* 16 (1988) 299–325.
- [6] J. Komlós, J. Pach, G. Woeginger, Almost tight bounds for epsilon-nets, preprint.
- [7] N. Sauer, On the density of families of sets, *J. Combin. Theory A* 13 (1972) 145–147.
- [8] V.N. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer, New York, 1982.
- [9] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, 1995.
- [10] V.N. Vapnik, A.Y. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, *Theory Probab. Appl.* 16 (2) (1971) 264–280.